CLAIMS

1. A processor system comprising:

a processor,

a first memory, being of a random access memory type,

a second memory, being of a random access memory type,

memory allocation means for allocation of data of a load module of said second memory to said first memory, said load module comprising variable/record data and/or instruction data,

an execution profiling section for providing execution data concerning behaviour of programs executed in the processor system, continuously or intermittently, whereby the operation of said means for memory allocation is software run-time updated based on said execution data, said execution profiling section in turn comprising at least one means for measuring the performance characteristics of part entities of said load module,

whereby said memory allocation means is arranged for allocation of selected part entities of said load module to said first memory.

2. The processor system according to claim 1, wherein at least one of said selected part entities is an individual variable/record, instructions of a program routine or instructions of a basic block, whereby basic block being defined as a straight sequence of instructions with the beginning and end as the sole enter and exit point, respectively.

3. The processor system according to claim 1, wherein said execution profiling section is arranged to select programs, for which said execution data is to be provided, according to internal information of an operating system.

4. The processor system according to claim 3, wherein said internal information comprises information about the priority of the program and/or whether the program is executed as a maintenance or background job.

5. The processor system according to claim 3, wherein said internal information comprises information about sizes of said part entities of said load module.

6. The processor system according to claim 1, wherein said execution profiling section comprises means for measuring the performance for a first type of data, variable/record data or instruction data, and in that said means for memory allocation comprises means for selecting part entities of a load module of both types of data as candidates for being allocated to said first memory, based on said measured performance of said first type of data.

7. The processor system according to claim 1, wherein said means for measuring the performance characteristics comprises means for measuring the number of accesses to part entities of said load module.

8. The processor system according to claim 7, wherein said means for measuring the performance characteristics comprises means for measuring the number of read accesses to part entities of said load module.

9. The processor system according to claim 1, wherein said means for measuring the performance characteristics comprises means for measuring the waiting time for access to part entities of said load module.

10. The processor system according to claim 7, wherein said means for measuring the performance characteristics comprises a hardware counter.

11. The processor system according to claim 10, wherein said execution profiling section further comprises a timer, for calibration of measured performance characteristics.

12. The processor system according to claim 9, wherein said means for measuring the performance characteristics comprises a hardware counter.

13. The processor system according to claim 12, wherein said execution profiling section further comprises a timer, for calibration of measured performance characteristics.

14. The processor system according to claim 7, wherein said means for measuring the number of accesses is implemented by code instrumentation.

15. The processor system according to claim 9, wherein said means for measuring the number of accesses is implemented by code instrumentation.

16. The processor system according to claim 7, wherein said means for measuring the number of accesses is implemented in an emulator or virtual machine.

17. The processor system according to claim 9, wherein said means for measuring the number of accesses is implemented in an emulator or virtual machine.

18. The processor system according claim 1, wherein said means for memory allocation is arranged to read a link table.

19. The processor system according to claim 18, wherein at least a part of said link table is allocated to said first memory.

20. The processor system according to claim 14, wherein said link table supports run-time linking.

21. The processor system according to claim 1, wherein said first memory is connected to said processor by a dedicated bus.

22. The processor system according to claim 1, wherein said first memory is implemented in a memory area located on the processor chip.

23. The processor system according to claim 1, wherein said first memory comprises at least one static random access memory.

24. The processor system according to claim 1, further comprising a third memory acting as a first level cache memory, whereby said first memory constitutes a second level cache memory.

25. The processor system according to claim 1, wherein said means for memory allocation operates according to a first algorithm for modifying said allocation data upon start, restart or software change of the processor system, and according to a second algorithm for modifying said allocation data at a later occasion.

26. The processor system according to claim 21, wherein said first algorithm is substantially based on information about sizes of part entities of said load module.

27. The processor system according to claim 1, wherein the data allocated to said first memory further comprises reference data.

28. The processor system according to claim 1, wherein said first memory is organised as a combined memory for at least two of the following types:

variables or records,
instructions, and
tables.

29. The processor system according to claim 1, wherein said first memory is organised as a split memory for the following types:

variables or records,
instructions, and
tables.

30. A method for memory handling in a processor system, comprising the steps of:

providing allocation data associated with a first set of data of a load module for allocation to a first memory, being of a random access memory type, from a second memory, being of a random access memory type, said load module comprising variable/record data and/or instruction data;

if said first set of data is allocated to a first memory, accessing said first memory for said first set of data;

providing, continuously or intermittently, execution data concerning behaviour of programs executed in said processor system, said step of providing execution data in turn comprising the step of measuring the performance characteristics of part entities of said load module;

modifying said allocation data by software in a run-time manner, based on said execution data; and

whereby said allocation data is arranged for allocation of selected part entities of said load module to said first memory.

31. The method according to claim 30, wherein at least one of said selected part entities is an individual variable/record, instructions of a program routine or instructions of a basic block, whereby basic block being defined as a straight sequence of instructions with the beginning and end as the sole enter and exit point, respectively.

32. The method according to claim 30, wherein said step of providing execution data provides execution data concerning behaviours of programs, selected according to internal information of an operating system.

33. The method according to claim 32, wherein said internal information comprises information about the priority of the program and/or if the program is executed as a maintenance or background job.

34. The method according to claim 32, wherein said internal information comprises information about sizes of said part entities of said load module.

35. The method according to claim 30, wherein said step of measuring the performance characteristics comprises the steps of:

measuring the performance for a first type of data, variable/record data or instruction data,

selecting part entities of a load module of both types of data as candidates for being allocated to said first memory, based on said measured performance of said first type of data.

36. The method according to claim 30, wherein said step of measuring the performance characteristics comprises the steps of measuring the number of accesses to part entities of said load module.

37. The method according to claim 36, wherein said step of measuring the performance characteristics comprises the steps of measuring the number of read accesses to part entities of said load module.

38. The method according to claim 30, wherein said step of measuring the performance characteristics comprises the steps of measuring the waiting time for access to part entities of said load module.

39. The method according to claim 30, wherein said step of providing execution data further comprises the steps of:

measuring the time period of said performance characteristics measurement; and

calculating a normalised rate from the measured performance characteristics and the measured time period.

40. The method according to claim 30, wherein said step of providing allocation data comprises the step of reading of link tables.

41. The method according to claim 40, wherein said link table supports run-time linking.

42. The method according to claim 30, wherein said step of modification favours allocation of data having the highest measured performance importance per time unit to said first memory.

43. The method according to claim 30, wherein the data allocated to said first memory further comprises reference data.

44. The method according to claim 30, further comprising the step of using a first algorithm for modifying said allocation data upon start or restart of the processor system, and switching to a second algorithm for modifying said allocation data at a later occasion.

45. The method according to claim 44, wherein said first algorithm is substantially based on information about sizes of part entities of said load module.

46. The method according to claim 30, further comprising the step of packing the content of a second memory from which data are re-allocated to said first memory, said packing occurring periodically and/or after larger re-allocations.

47. The method according to claim 30, further comprising the step of run-time updating of said software controlling said modification of said allocation data.

48. A processor system comprising:

a processor,

a first memory being of a random access memory type,

a second memory being of a random access memory type,

memory allocation means for allocation of data of a load module of said second memory to said first memory, said load module comprising variable/record data and/or instruction data, said memory allocation section being arranged for allocation of selected part entities of said load module to said first memory, said selection being based on internal information of an operating system.

49. The processor system according to claim 48, wherein at least one of said selected part entities is an individual variable/record, instructions of a program routine or instructions of a basic block, whereby basic block being defined as a straight sequence of instructions with the beginning and end as the sole enter and exit point, respectively.

50. The processor system according to claim 48, wherein said internal information comprises information about the priority of the program and/or information of whether the program is executed as a maintenance or background job.

51. The processor system according to claim 48, wherein said internal information comprises information about sizes of said part entities of said load module.

52. A method for memory handling in a processor system, comprising the steps of:

allocating data associated with a first set of data of a load module into a first memory, being of a random access memory type, from a second memory, being of a random access memory type, said load module comprising variable/record data and/or instruction data, said step of allocating data in turn comprising the step of providing internal information of an operating system regarding part entities of said load module, whereby the allocation of data is performed on selected part entities; and

accessing said first memory for said first set of data.

53. The method according to claim 52, wherein at least one of said selected part entities is an individual variable/record, instructions of a program routine or instructions of a basic block, whereby basic block being

defined as a straight sequence of instructions with the beginning and end as the sole enter and exit point, respectively.

54. The method according to claim 52, wherein said internal information comprises information about the priority of the program and/or if the program is executed as a maintenance or background job.

55. The method according to claim 52, wherein said internal information comprises information about sizes of said part entities of said load module.
